The following document contains information on Cypress products. Although the document is marked with the name "Spansion" and "Fujitsu", the company that originally developed the specification, Cypress will continue to offer these products to new and existing customers.

Continuity of Specifications
There is no change to this document as a result of offering the device as a Cypress product. Any changes that have been made are the result of normal document improvements and are noted in the document history page, where supported. Future revisions will occur when appropriate, and changes will be noted in a document history page.

Continuity of Ordering Part Numbers
Cypress continues to support existing part numbers. To order these products, please use only the Ordering Part Numbers listed in this document.

For More Information
Please contact your local sales office for additional information about Cypress products and solutions.

About Cypress
Cypress (NASDAQ: CY) delivers high-performance, high-quality solutions at the heart of today's most advanced embedded systems, from automotive, industrial and networking platforms to highly interactive consumer and mobile devices. With a broad, differentiated product portfolio that includes NOR flash memories, F-RAM™ and SRAM, Traveo™ microcontrollers, the industry's only PSoC® programmable system-on-chip solutions, analog and PMIC Power Management ICs, CapSense® capacitive touch-sensing controllers, and Wireless BLE Bluetooth® Low-Energy and USB connectivity solutions, Cypress is committed to providing its customers worldwide with consistent innovation, best-in-class support and exceptional system value.

Colophon
The products described in this document are designed, developed and manufactured as contemplated for general use, including without limitation, ordinary industrial use, general office use, personal use, and household use, but are not designed, developed and manufactured as contemplated (1) for any use that includes fatal risks or dangers that, unless extremely high safety is secured, could have a serious effect to the public, and could lead directly to death, personal injury, severe physical damage or other loss (i.e., nuclear reaction control in nuclear facility, aircraft flight control, air traffic control, mass transport control, medical life support system, missile launch control in weapon system), or (2) for any use where chance of failure is intolerable (i.e., submersible repeater and artificial satellite). Please note that Spansion will not be liable to you and/or any third party for any claims or damages arising in connection with above-mentioned uses of the products. Any semiconductor devices have an inherent chance of failure. You must protect against injury, damage or loss from such failures by incorporating safety design measures into your facility and equipment such as redundancy, fire protection, and prevention of over-current levels and other abnormal operating conditions. If any products described in this document represent goods or technologies subject to certain restrictions on export under the Foreign Exchange and Foreign Trade Law of Japan, the US Export Administration Regulations or the applicable laws of any other country, the prior authorization by the respective government entity will be required for export of those products.

Trademarks and Notice
The contents of this document are subject to change without notice. This document may contain information on a Spansion product under development by Spansion. Spansion reserves the right to change or discontinue work on any product without notice. The information in this document is provided as is without warranty or guarantee of any kind as to its accuracy, completeness, operability, fitness for particular purpose, merchantability, non-infringement of third-party rights, or any other warranty, express, implied, or statutory. Spansion assumes no liability for any damages of any kind arising out of the use of the information in this document.

MCU-AN-510042-E-10

FM3

32-BIT MICROCONTROLLER

MB9A310 Series

FSS MB9AF314L EV-BOARD

USER MANUAL

APPLICATION NOTE

For more information for the FM3 microcontroller, visit the web site at:
http://www.fujitsu.com/global/services/microelectronics/product/micom/roadmap/industrial/fm3/

Revision History

Date	Author	Version	Remark
2012-04-20	Edison Zhang	V1.0	

Table of Contents

1 Introduction

1.1 Product Overview

FSS MB9AF314L EV-Board (PN: FSSDC-9A314-EVB) provides an economical and simple means for study usage for MB9A310 series MCU. The board compatible with both 3.3 and 5V system contains some external resources (LED, Key, USB Device, USB Host interface…) to demonstrate MCU periphery function.

It also provides standard 20 pin JTAG interface, which is both compatible with IAR and Keil debug tool. In addition, it allows On-board programming with USB mode.

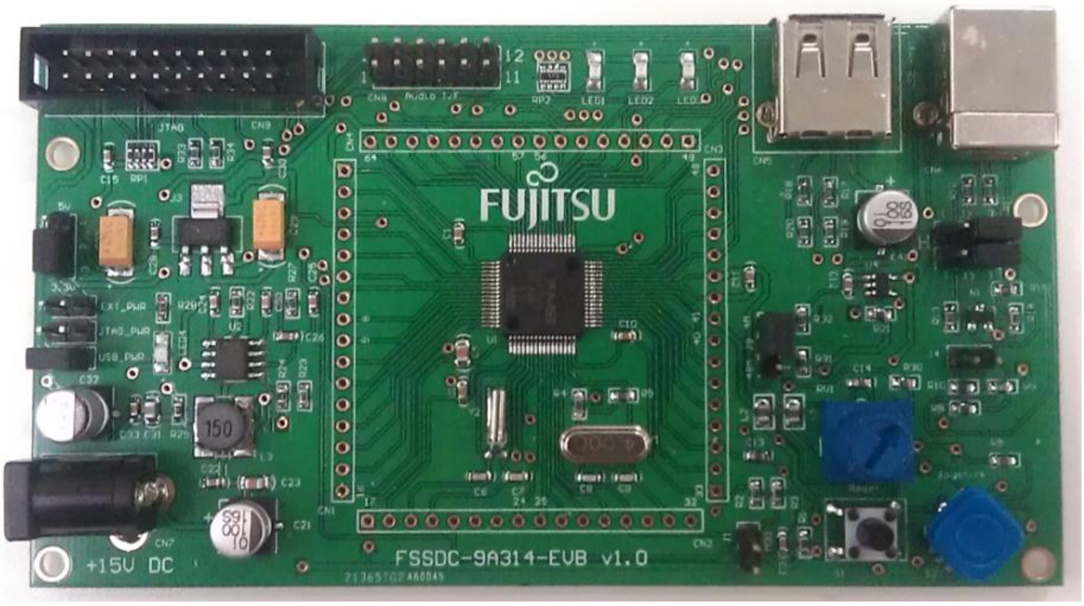


Figure 1-1: FSSDC-9A314-EVB Overview

1.2 MB9A310 Series MCU

MB9A310 series MCU is 32-bit general purpose MCU of FM3 family that features the industry's leading-edge ARM Cortex-M3TM CPU and integrates Fujitsu's highly reliable and high-speed secure embedded flash technology. This MCU can operate at up to 40MHz CPU frequency and work at a wide voltage range (2.7V-5.5V), which can be both compatible with 3.3V and 5V system.

It includes a host of robust peripheral features, including motor control timers (MFT), base timer (can be configured to PWM, PPG, Reload, PWC timer), ADCs, on-chip memory (up to 256K Flash, up to 32K SRAM) and a wide range of communication interfaces (USB, I2C, SIO, LIN, CAN).

The size of on-chip memory can be configured according to different part number and the package is available in LQFP and BGA, shown in following table.

Product	Flash	SRAM	Package
MB9AF316M/N	512kB	32kB	M: LQFP-80 N: LQFP-100
MB9AF315M/N	384kB	32kB	M: LQFP-80 N: LQFP-100
MB9AF314L/M/N	256kB	32kB	L: LQFP-64 M: LQFP-80 N: LQFP-100
MB9AF312L/M/N	128kB	16kB	L: LQFP-64 M: LQFP-80 N: LQFP-100
MB9AF311L/M/N	64kB	16kB	L: LQFP-64 M: LQFP-80 N: LQFP-100

Table 1-1: Product List

1.3 Board Features

FSS MB9AF314L EV-Board includes following features:

- Microcontroller MB9AF314L
- 1x USB-Host (Type-A connector)
- 1x USB-Device (Type-B connector)
- Standard JTAG Interface on a 20 pin-header
- 3x LED
- 1x Joystick
- 1x ‘Reset’-button
- 1x potentiometer regulating input voltage to AD channel
- All 64 pins routed to test pads
- Audio Interface
- On-board 5V and 3V voltage regulators, ‘Power’-LED
- Power supply via USB-Device, JTAG or external with 15V power connector

1.4 Getting Started

The microcontroller on the FSSDC-9A314-EVB is already programmed with a test program. Please follow the test process as below first.

1) Make sure J7 is short, and connect the FSSDC-9A314-EVB with PC via USB Function interface (Type-B Connector)

2) Watch that the LED3 blinks for a short time, then stays at “ON” state; LED1 and LED2 plays as horse-race lamp.

3) Turning the potentiometer clockwise will make the LED1 and LED2 blink fast, and turning the potentiometer anti-clockwise will make the LED1 and LED2 blink slowly.

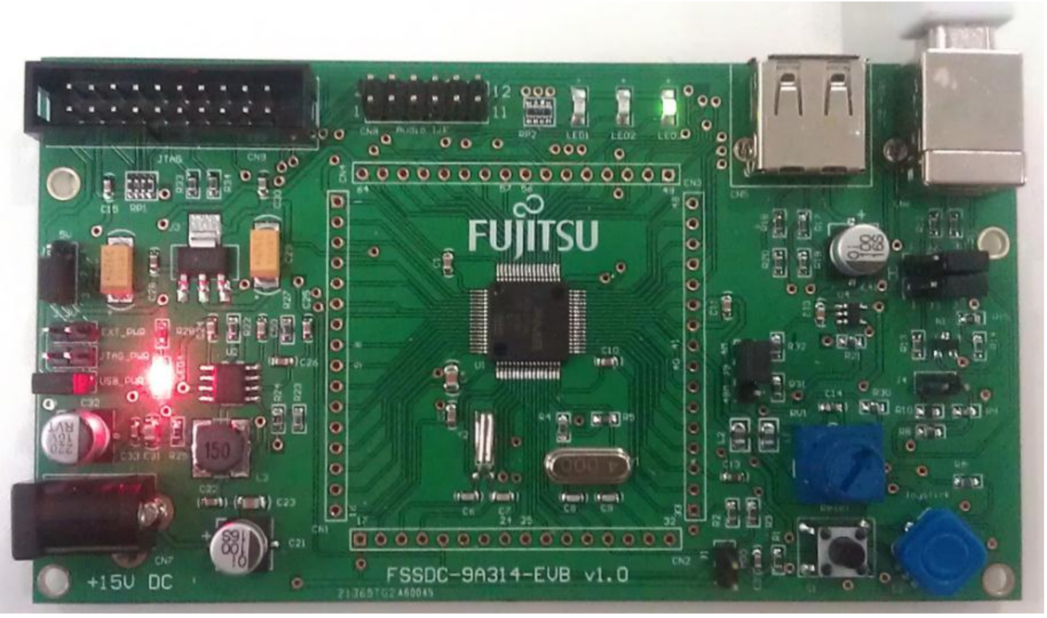


Figure 1-2: FSSDC-9A314-EVB Test

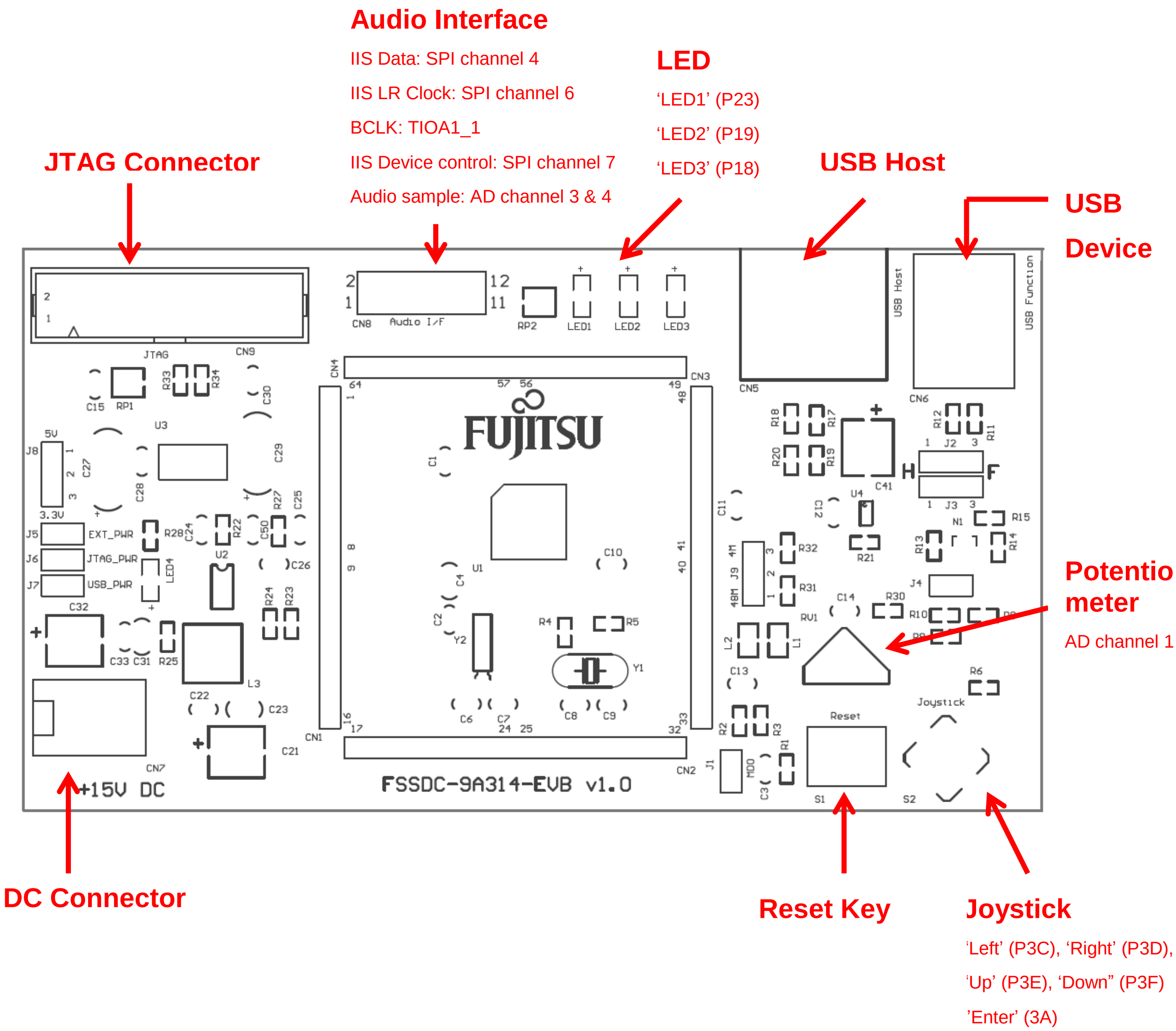
4) Actually now, the FSSDC-9A314-EVB can be used as a simple USB mouse, press the up/down/left/right key of Joystick to move the cursor on the screen, and the enter key of Joystick acts as “Left Click” of mouse.

If the test result is normal as described, congratulation! You finished the first test on the FSSDC-9A314-EVB, and now you will get more details about FSSDC-9A314-EVB at following chapters. You will learn about:

- ***The hardware setting***
- ***How to program the Flash***
- ***How to start with IAR-Embedded-Workbench and Keil µVision***

2 Hardware Setting

2.1 Main Features



2.2 Jumpers Overview

Jumper Name	Function	Setting
J5-J7	Power selection	Short J5: select external power Short J6: select JTAG power Short J7: select USB power
J8[*2]	Voltage selection	Short 1,2: 5V Short 2,3: 3.3V
J2 J3	USB Host/Function selection	Short 1,2: USB Host Short 2,3: USB Function
J1	Mode setting	Short: Programming mode Open: Normal mode
J9	Oscillator Selection	Short 1,2: when 48M oscillator is used Short 2,3: when 4M oscillator is used
J4	Voltage division	Short: when 5V power is used Open: when 3.3V power is used

Table 2-1: Jumper List

Attention:

1) **Never short J5-J7 at the same time! When Keil U-Link ME is used, JTAG power can not be used.**
2) **When Keil U-link is used, only 3.3V can be selected, never select 5V power!**

2.3 Setting for USB On-Board Programming

Jumper Number	Function	Setting
CN6	USB function connector	Connect with PC via USB cable
J2 J3	Host/Function selection	Short 2,3
J1	Mode setting	Short
J9	Oscillator Selection	Short 2,3
J4	Voltage division	Short: when 5V power is used Open: when 3.3V power is used

Table 2-2: Setting for USB On-Board Programming

2.4 Setting for Debug Tool

■ Use IAR J-Link

Part Number	Function	Setting
CN9	JTAG connctor	Connect with J-Link
J5-J7	Power selection	Short J5: select external power Short J6: select JTAG power Short J7: select USB power
J8	Voltage selection	Short 1,2: 5V Short 2,3: 3.3V
J1	Mode setting	Open

Table 2-3: Setting for J-Link

■ Use Keil U-Link ME

Part Number	Function	Setting
CN9	JTAG connctor	Connect with U-Link ME
J5-J7	Power selection	Short J5: select external power Short J7: select USB power
J8	Voltage selection	Short 2,3
J1	Mode setting	Open

Table 2-4: Setting for U-Link

3 Flash On-Board Programming

3.1 On-Board Programming via USB

- First check the hardware setting as introduced by section 2.3.
- After connect with PC via USB cable, the EVB can be identified as a USB device.



Figure 3-1: USB Device Sign

- Check the COM port for this USB port in the device manager.

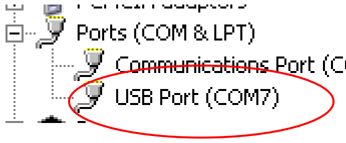


Figure 3-2: COM Port in Device Manager

- Install the USB programmer: usbdirect-v01l04.zip. (It can be downloaded on the web)
- Open it, set the parameter as shown in following figure, and select Hex file.

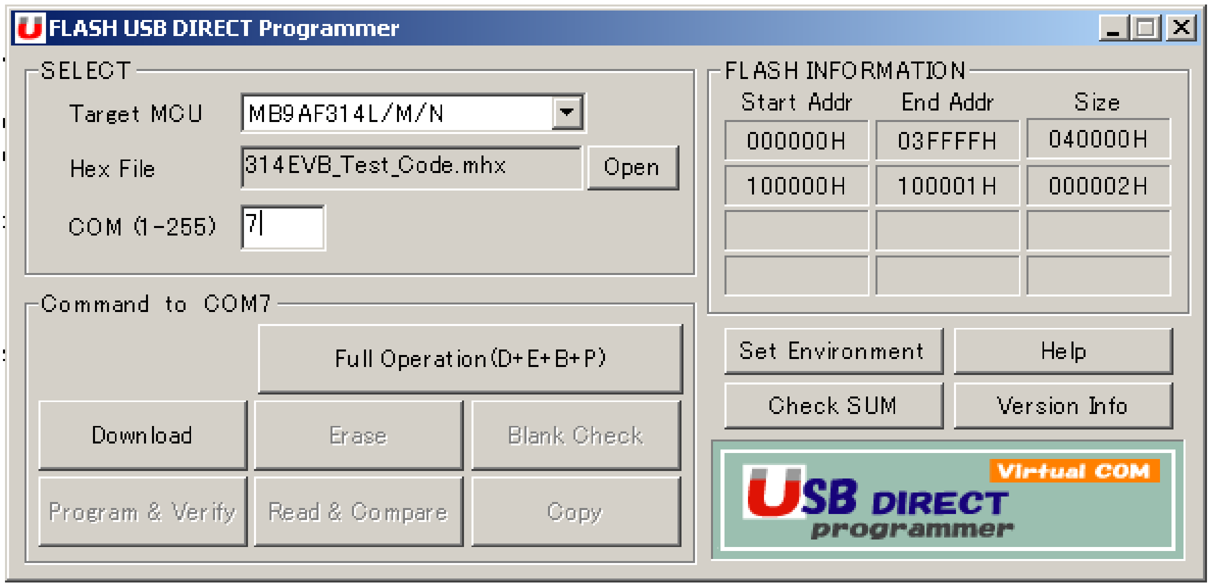


Figure 3-3: USB Programmer Overview

- Click Full Operation.

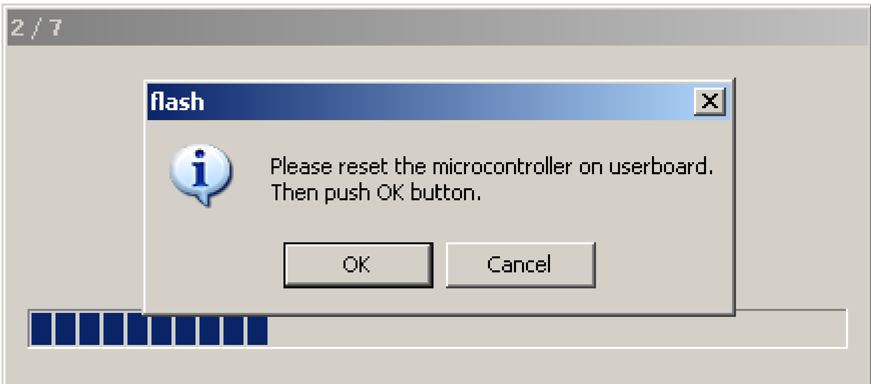


Figure 3-4: Press Reset Button

■ Press reset key on the board. Programming will start.

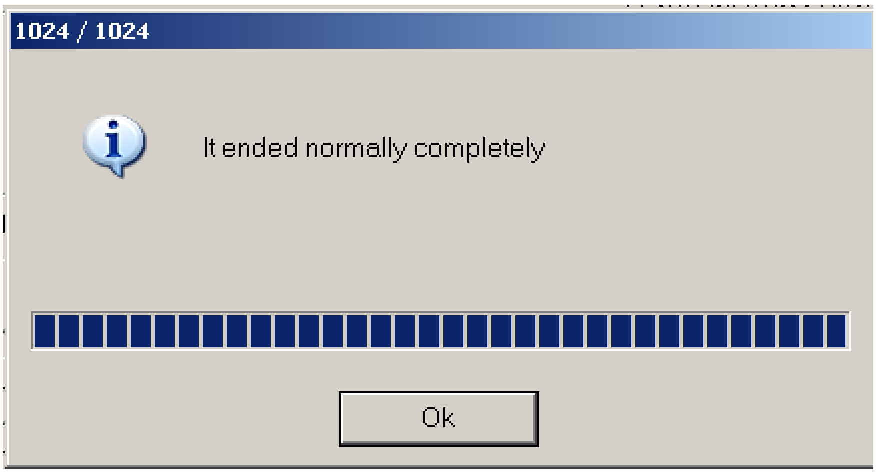


Figure 3-5: Start Programming

4 Sample Code

The sample codes for FSSDC-9A314-EVB board are listed as following table.

Project Name	Description
Blinky	Turn the potentiometer to change the LED blinking frequency.
usb_function_mouse	Implement the USB mouse to demonstrate USB function
usb_host_catch_mouse	Identify a USB mouse and print the mouse position on debug window
usb_host_msc	Demo USB Host mass storage class function, and implement FatFS in the sample code, user can create/write/read/delete files on the U-disk.

Notes:

1) **It provides both IAR and Keil project for these sample code, IAR project is developed in EWARM Embedded Workbench V6.21, and Keil project is developed in Keil uVision 4.21.**

2) **If user use other version to open these projects, compiling error may occur, in this case, please check following setting.**

- **IAR IDE**
  - **MCU type**
  - **Pre-included file**
  - **ICF file**
  - **Flash loader**

- **Keil IDE**
  - **MCU type**
  - **Pre-included file**
  - **ROM & RAM memory address**

4.1 Blinky

■ Hardware Setting

None

■ Flowchart

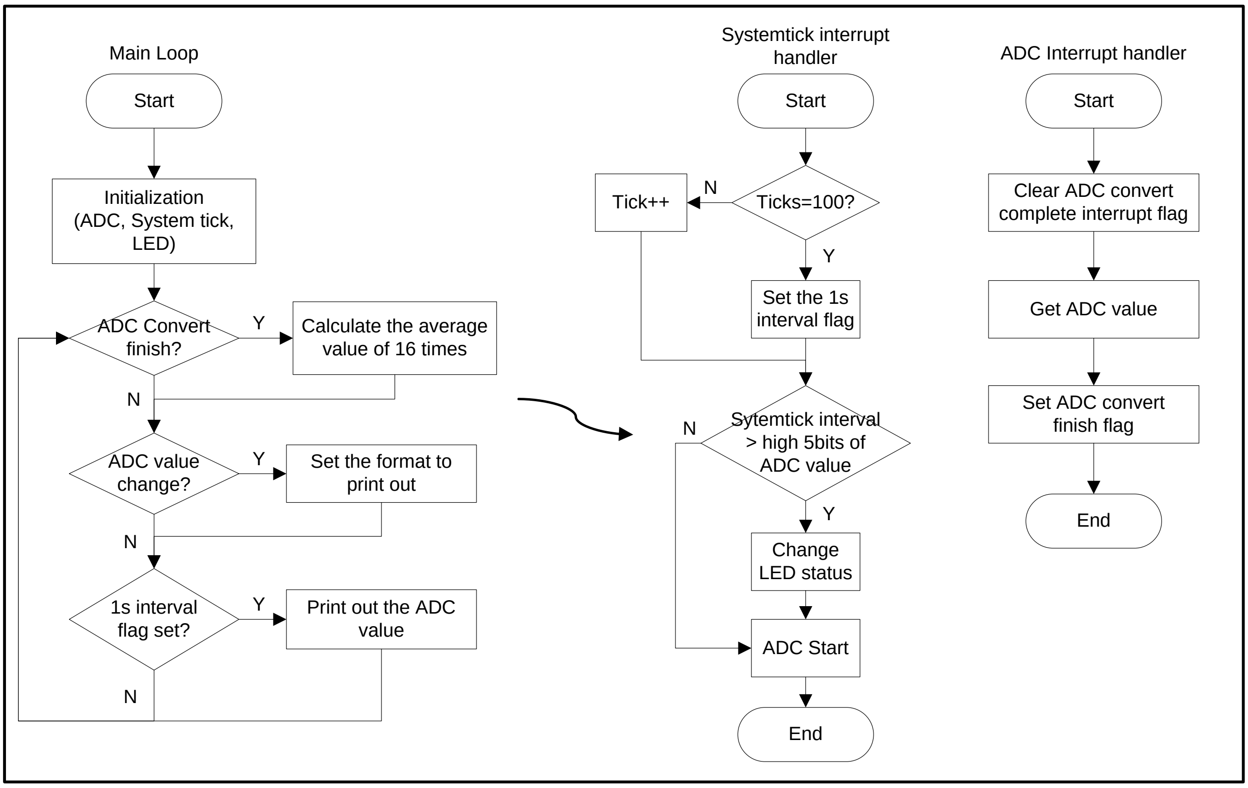


Figure 4-1: Blinky Flowchart

■ Usage

1) Turn the potentiometer (RV1) clockwise and anti-clockwise
2) Watch the LED blinking frequency

4.2 USB Function

■ Hardware Setting

- Check if 2,3 of J2 and J3 short
- Check J4 (Short: 5V, Open: 3.3V)
- Connect with PC via USB cable

■ Flowchart

The following flowchart illuminates the procedure to implement a USB mouse, it is not a certain flow of a function, but provides a clue to study and understand the sample code.

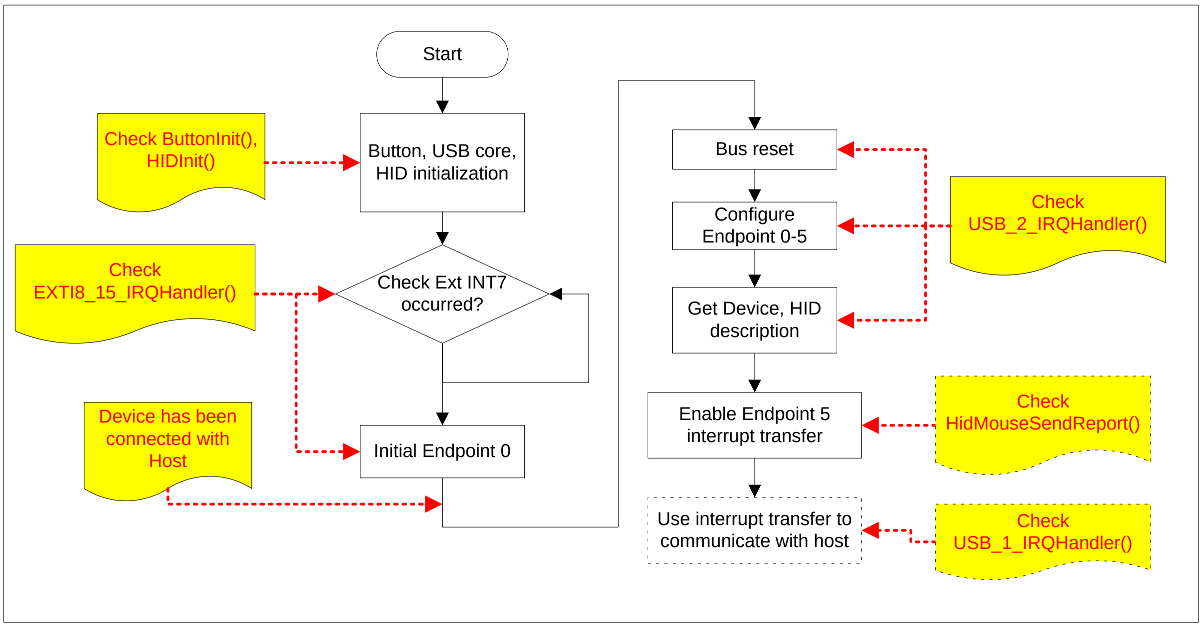


Figure 4-2: USB Function Sample Code Flowchart

■ Usage

1) Make the hardware setting.

2) Watch the mouse moving when pressing “Up”, “Down”, “Left”, “Right”, “Enter” keys of Joystick.

4.3 USB Host (Catch Mouse)

■ Hardware Setting

- ➢ Check if 1,2 of J2 and J3 short
- ➢ Connect with a USB mouse

■ Flowchart

The following flowchart illuminates the procedure to implement USB host function, which can catch the position of a USB mouse, it is not a certain flow of a function, but provides a clue to study and understand the sample code.

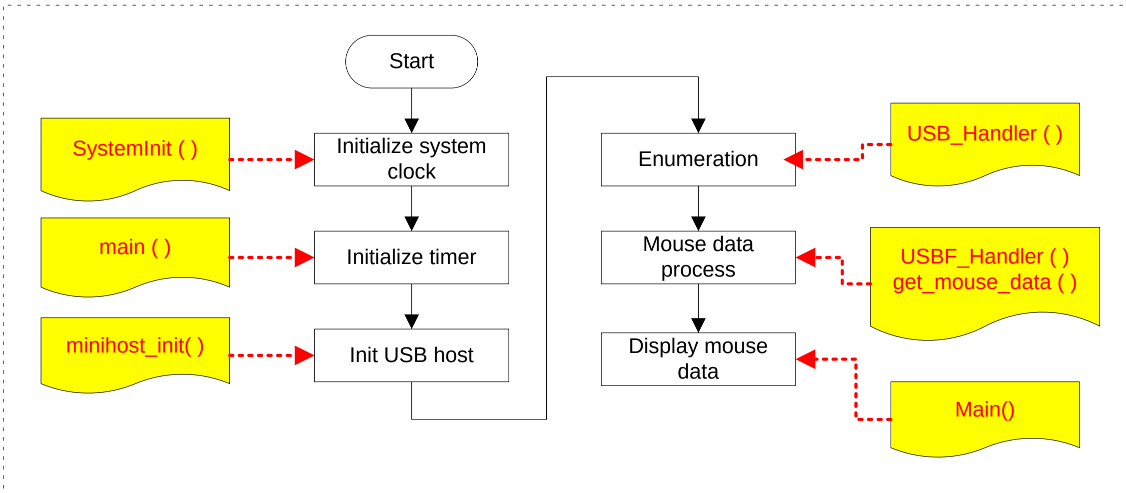


Figure 4-3: USB Host Catch Mouse Flowchart

■ Usage

1) Make the hardware setting
2) Enable definition "Debug" in the code
3) Run the code.
4) Move the USB mouse, and watch the mouse position in the terminal IO window as following figure. (Debug viewer window in Keil IDE)

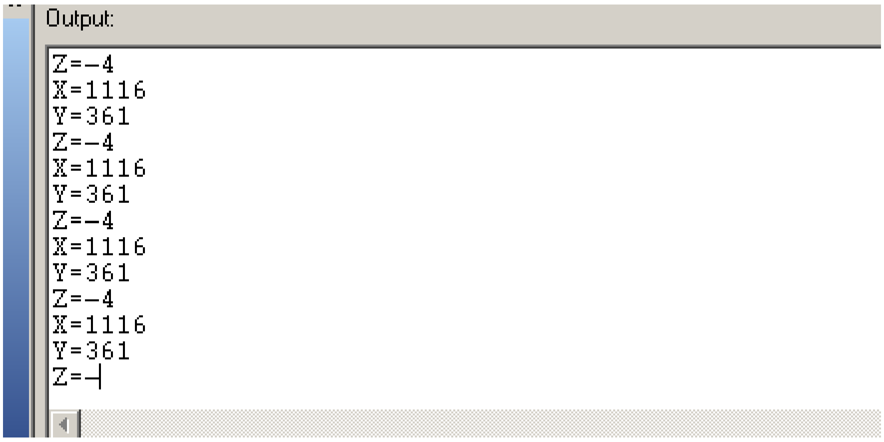


Figure 4-4: USB Mouse Position Display

4.4 USB Host (Access to U-Disk)

■ Hardware Setting

- ➢ Check if 1,2 of J2 and J3 short
- ➢ Connect with a U-Disk

■ Flowchart

The following flowchart illuminates the procedure to implement USB Host Mass Storage Class function, which can identify the U-Disk and make file operation on the U-Disk, it is not a certain flow of a function, but provides a clue to study and understand the sample code.

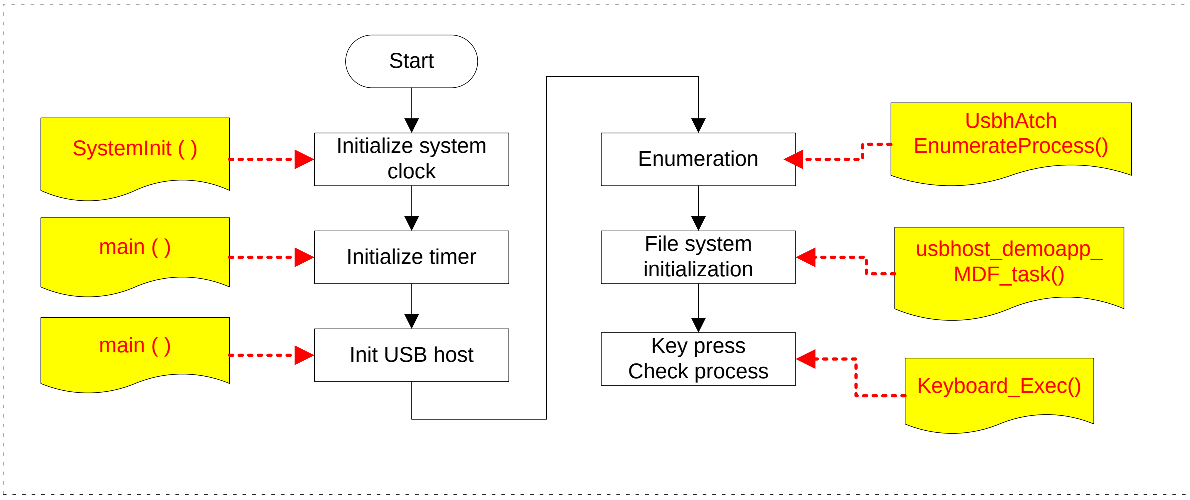


Figure 4-5: U-Disk Access Demo Flowchart

■ Usage

1) Run the code
2) Plug-in the U-Disk
3) Press “Left key” of Joystick
4) Plug-out the U-Disk and check if a file is created (Path: ..\F1\F2\) on PC.
5) Plug-in the U-Disk again
6) Press “Right key” of Joystick
7) Plug-out the U-Disk and check if the file is deleted

5 Debug Tool and IDE

FSS MB9AF314L EV-Board supports both Keil U-Link-ME and IAR J-Link for debug shown as following.



Figure 5-1: J-Link Overview

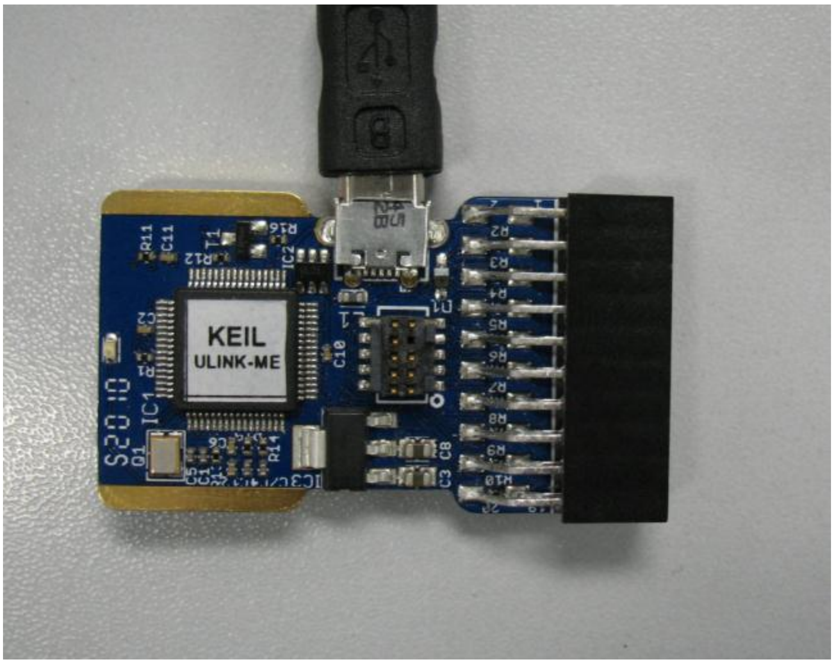


Figure 5-2: U-Link Overview

The U-Link-me should be used with Keil uVision 4 which can be downloaded freely from following web.

https://www.keil.com/update/sw/RVMDK/4.21

The J-Link should be used with IAR Embedded Workbench which can be downloaded freely from following web.

http://www.iar.com/website1/1.0.1.0/68/1/

5.1 Debug with J-Link in IAR EWARM Workbench

The sample code can be debugged in IAR EWARM Workbench with J-Link. The following figure shows basic debug window.

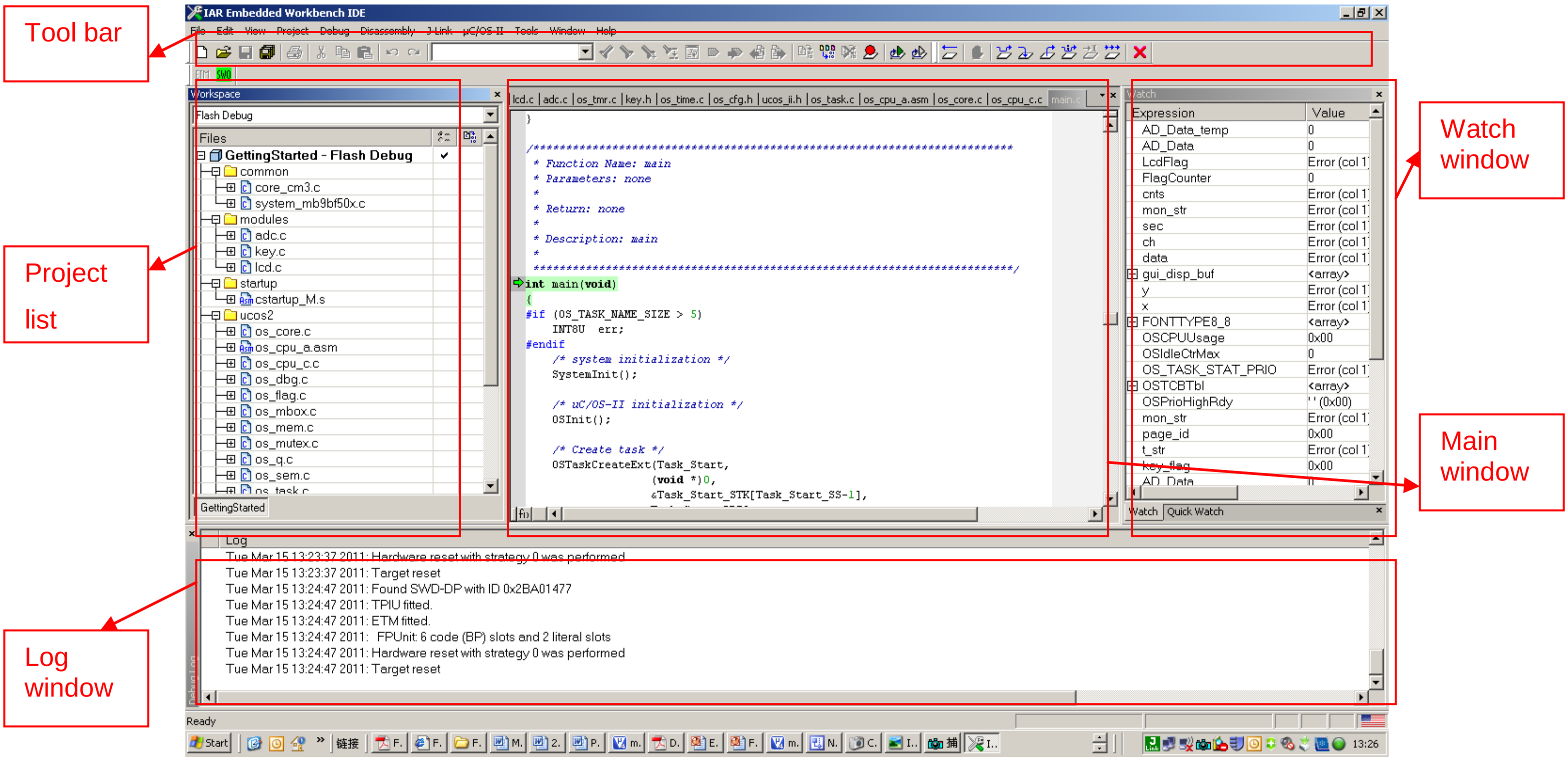


Figure 5-3: IAR IDE Overview

- **Run an Existed Project**

1) Open a project by clicking "File | Open | Workspace"

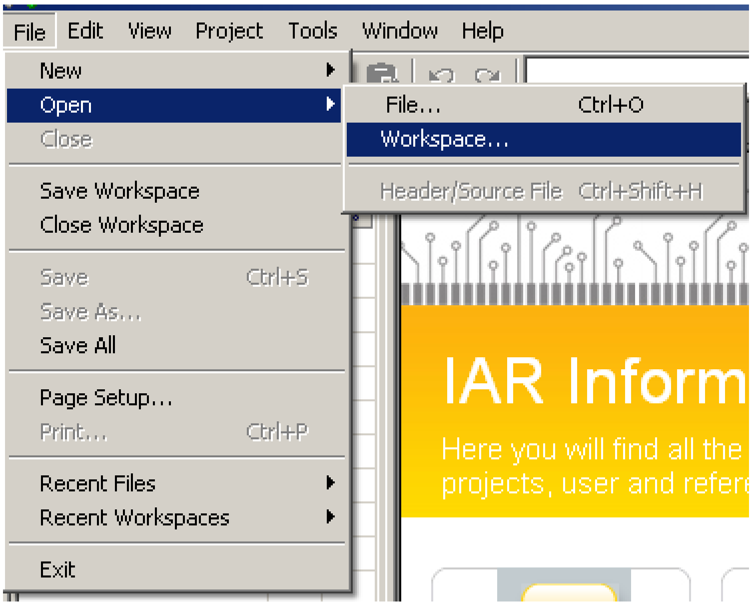


Figure 5-4: Open a Project

2) Select a project (eww file)

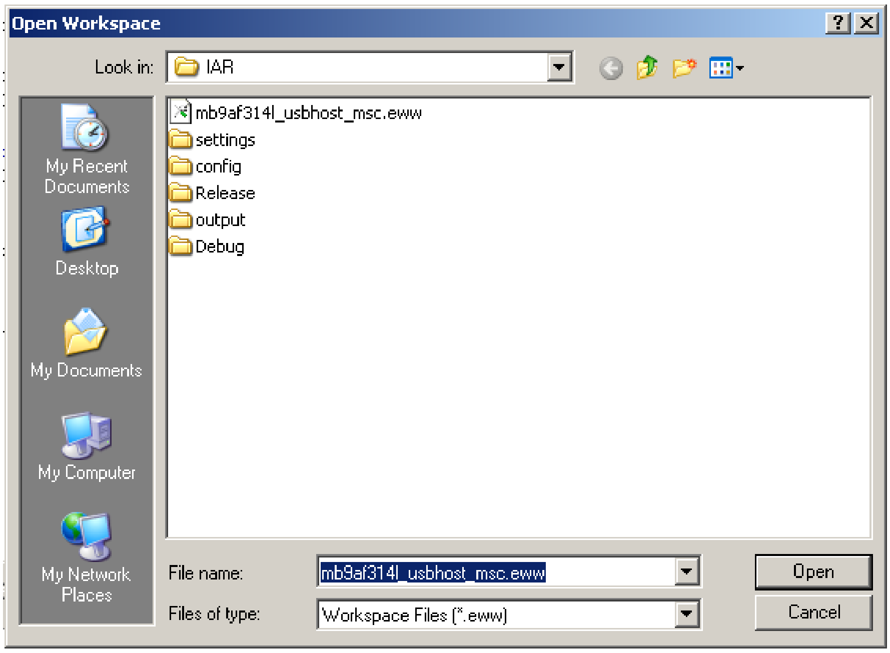


Figure 5-5: Select a Project

3) Click “Project | Rebuild All”

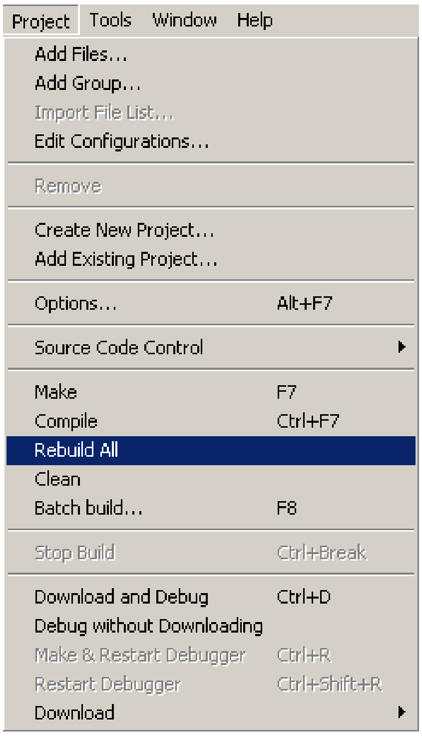


Figure 5-6:Rebuild All

4) Click "Download and Debug"



Figure 5-7:Click Download and Debug

5) Use following tool bar to debug

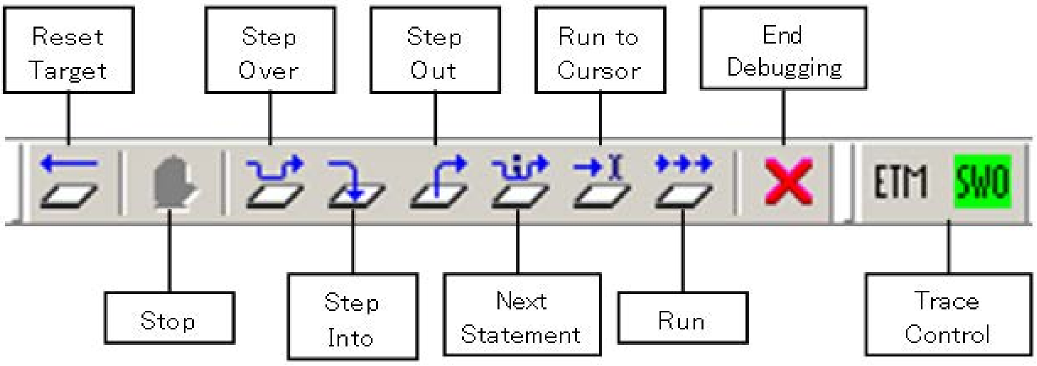


Figure 5-8: Debug Tool Bar

The sample codes support both Flash and RAM debug in IAR EWARM Workbench, if Flash debug is used, the code is programmed into MB9AF314L Flash. If RAM debug is used, the code only runs in RAM area, and after power off, the code will not be stored, but the RAM debug will be faster than Flash debug.

- **Setting for Flash Debug**

1) Check the configuration file path ($PROJ_DIR$\config\mb9af314.icf) in Linker table.

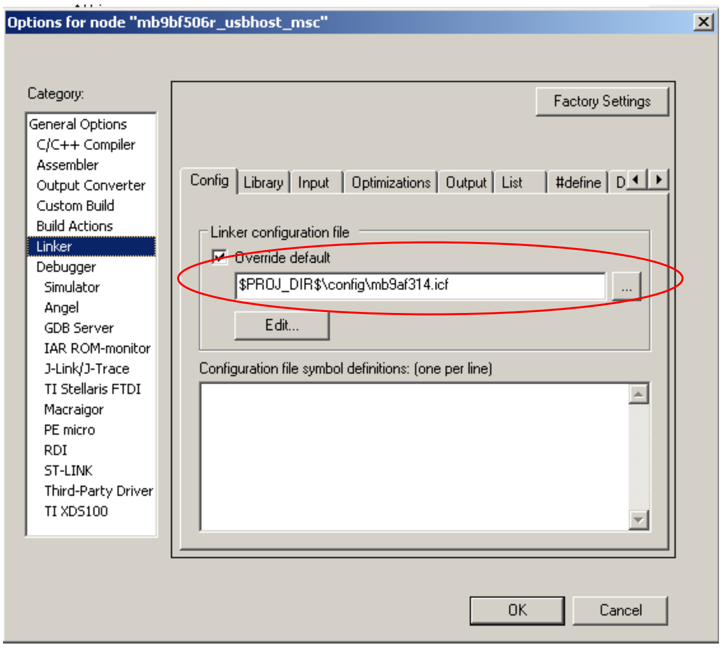


Figure 5-9: ICF File for Flash Debug

2) Don't select "Use macro files" in "Debugger|Setup" table.

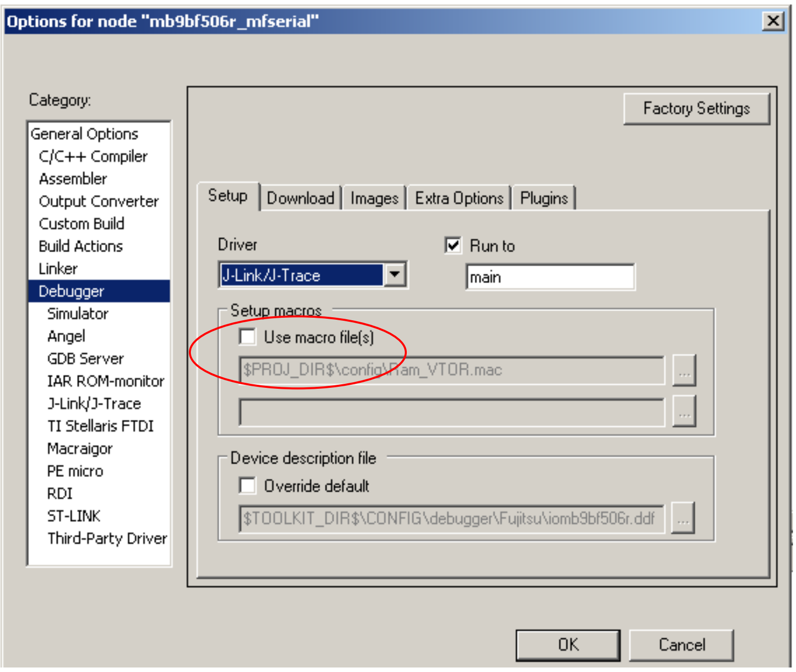


Figure 5-10: Macro File Disabled

3) Set Flash loader file path ($PROJ_DIR$\config\flashloader\FlashLoader.board) in "Debug|Download" table.

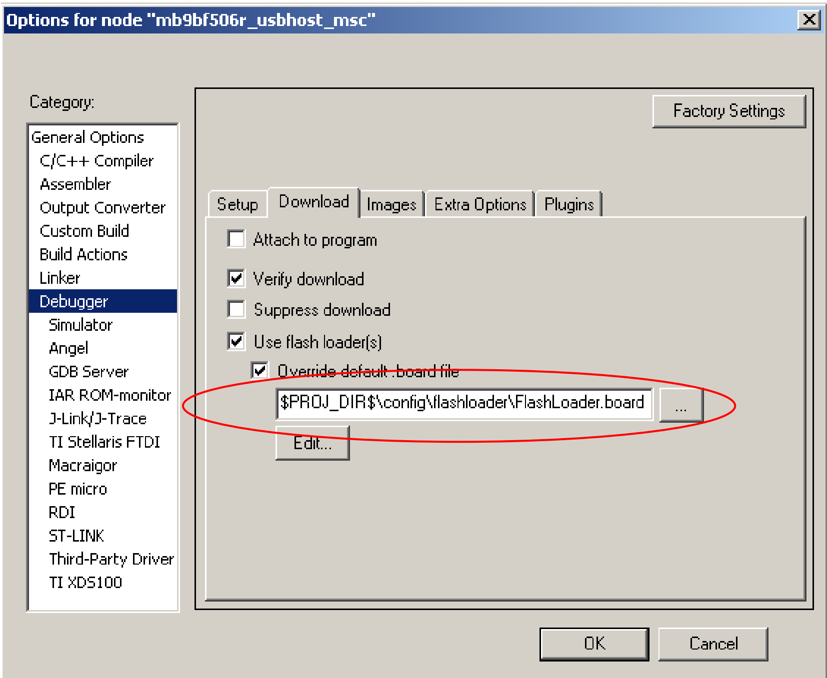


Figure 5-11: Flash Load File Path

■ **Setting for RAM Debug**

1) Check the configuration file path ($PROJ_DIR$\config\mb9bf314_ram.icf) in Linker table.

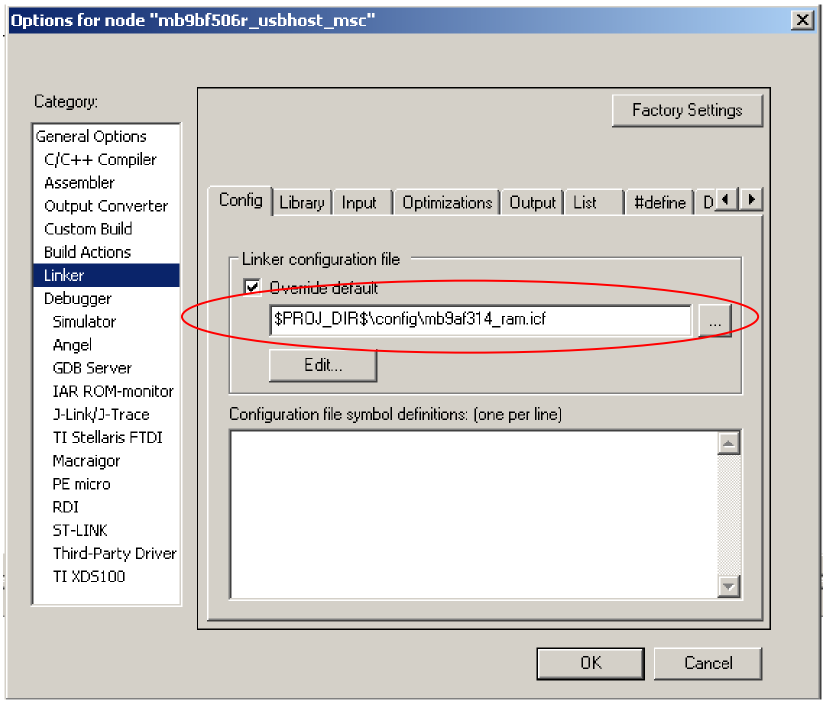


Figure 5-12: ICF File for RAM Debug

2) Select “Use macro files” in “Debugger|Setup” table.

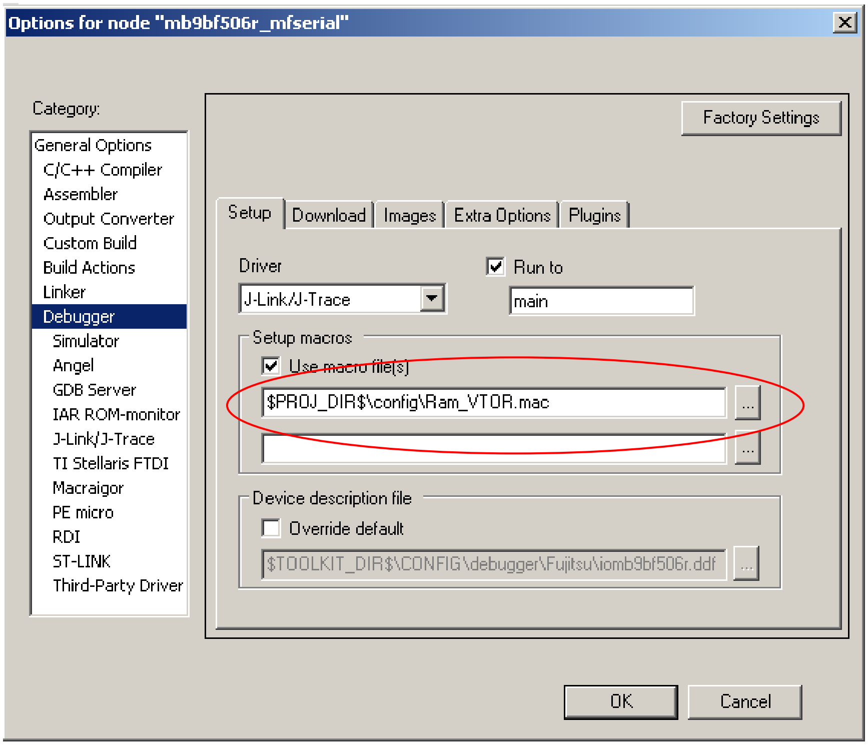


Figure 5-13: Macro File Enabled

3) Don’t use Flash loader file.

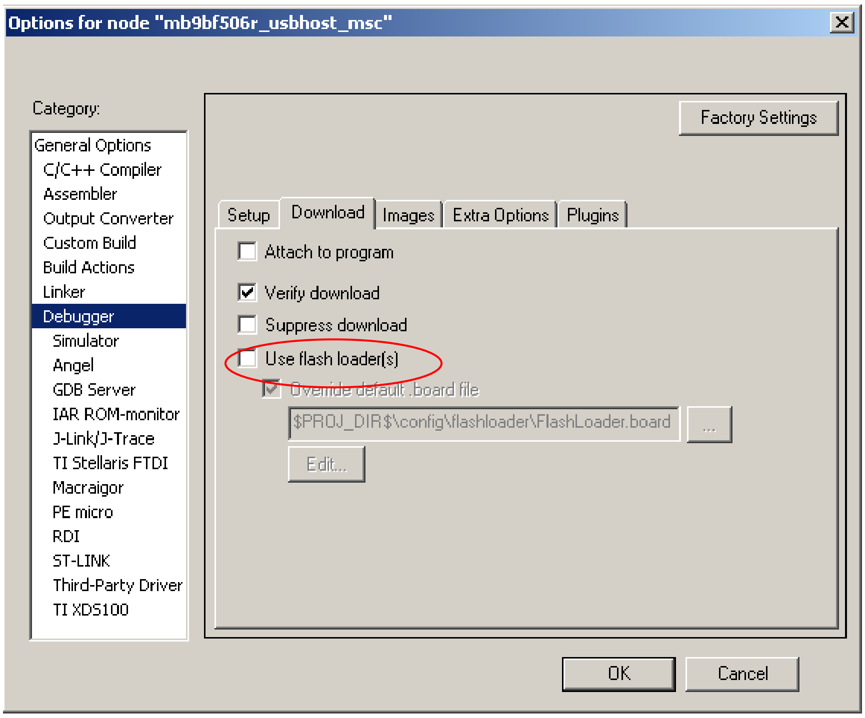


Figure 5-14: Flash Loader File Disabled

If user need to program the hex file into Flash via USB programmer, a hex file need to be produced first.

■ How to Make a HEX File

1) Use Flash debug

2) Select “Generate additional output” in “Output Converter” table.

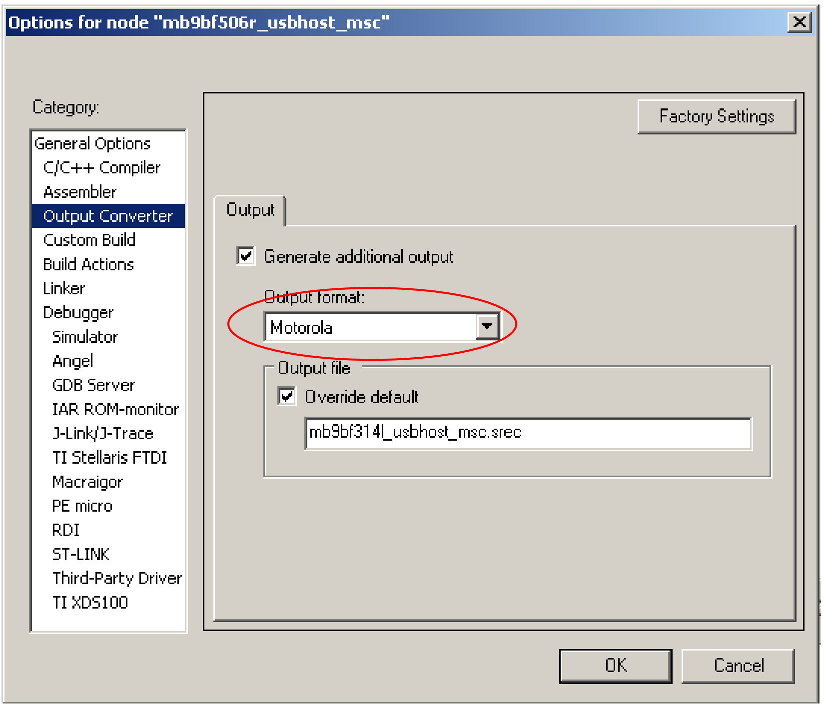


Figure 5-15: Hex File Genration in IAR IDE

3) User can find the generated file in path (../Debug/Exe)

5.2 Debug with U-Link ME in Keil uVision4

The sample code can also be debugged in Keil µVersion4 with U-Link. The following figure shows basic debug window.

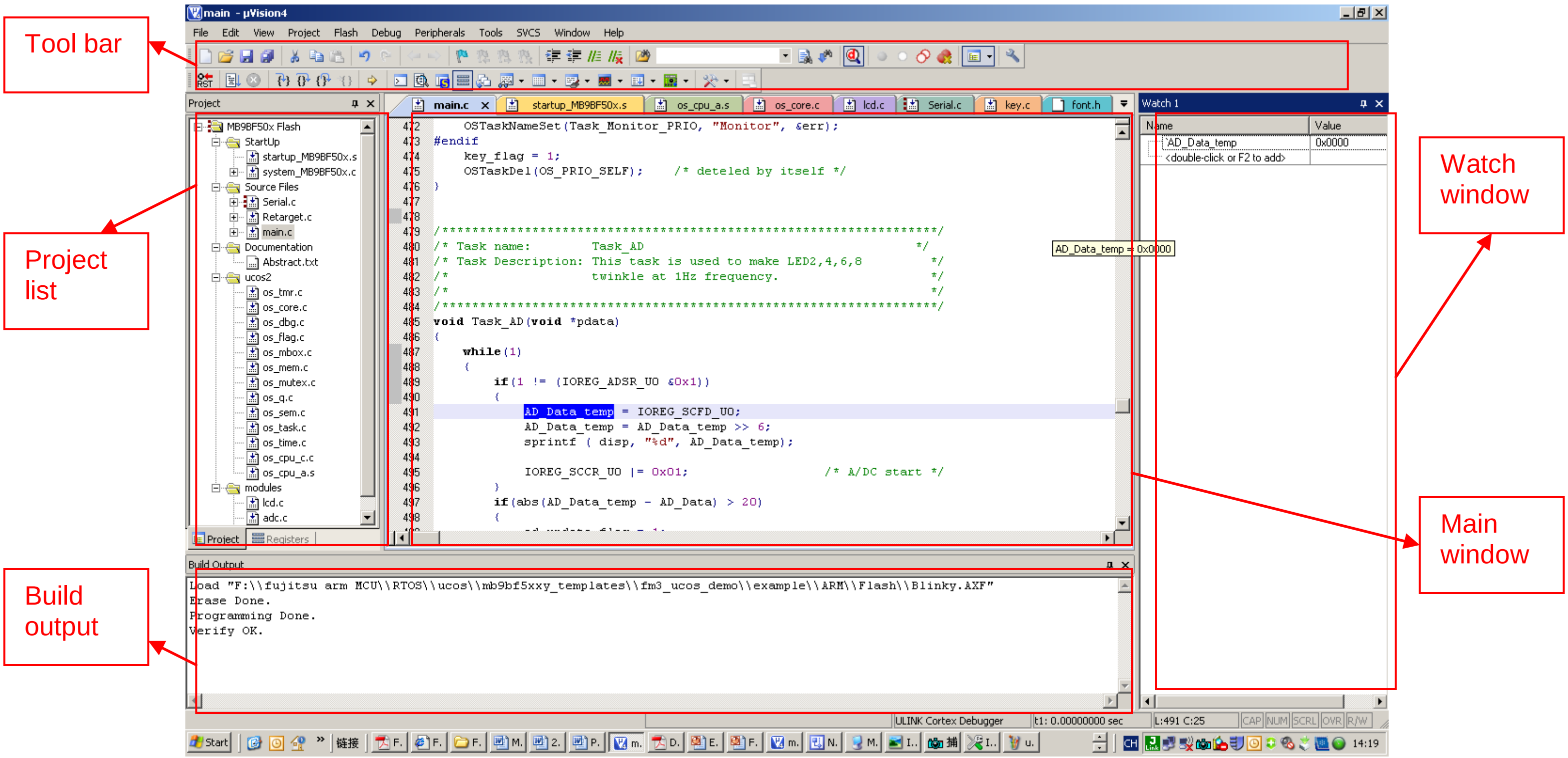


Figure 5-16: Keil IDE Overview

- **Run an Existed Project**

1) Open a project by clicking "Project | Open Project…"

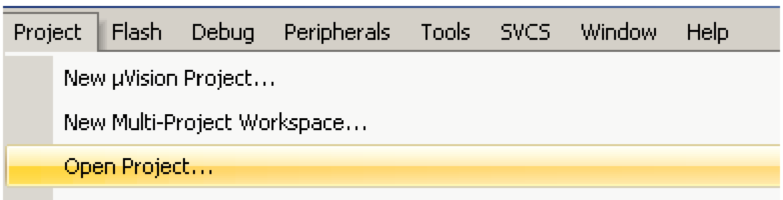


Figure 5-17: Open a Project

2) Select a project (uvproj file)

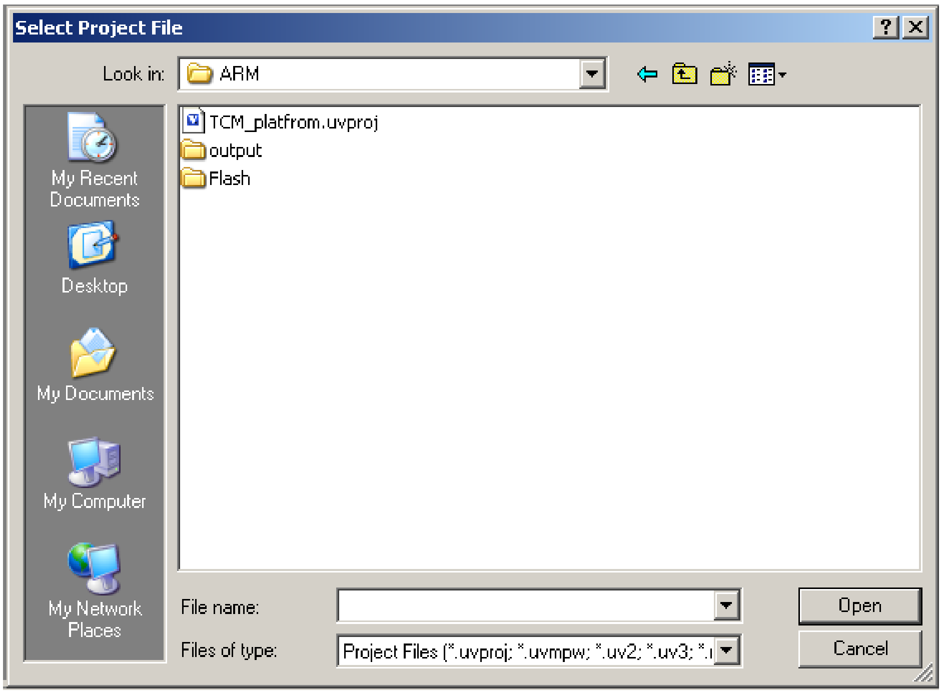


Figure 5-18: Select a Project

3) Rebuild all



Figure 5-19: Rebuild All Files

4) Start debug



Figure 5-20: Start Debug

5) Use following tool bar to debug

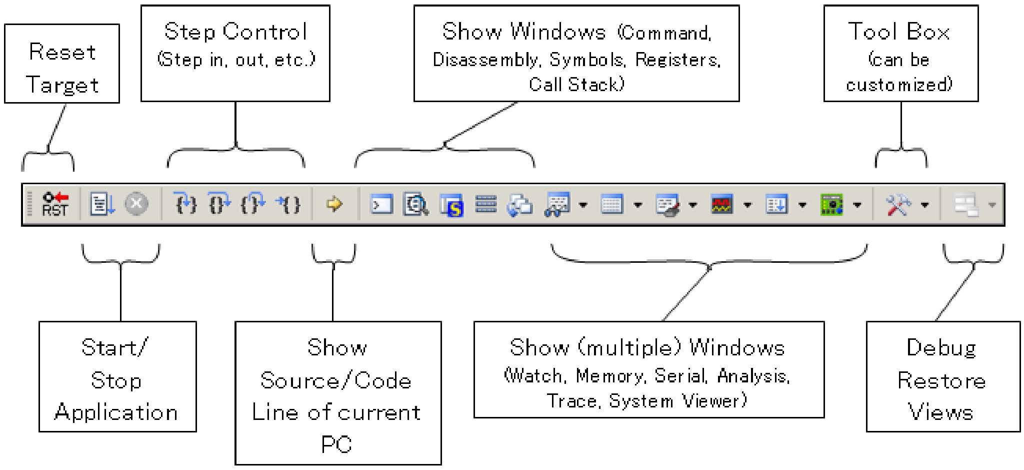


Figure 5-21: Debug Tool Bar

The sample codes support both Flash and RAM debug in Keil uVison 4, if Flash debug is used, the code is programmed into MB9AF314L Flash. If RAM debug is used, the code only runs in RAM area, and after power off, the code will not be stored, but the RAM debug will be faster than Flash debug.

■ Setting for Flash Debug

1) Set ROM address in Flash area. (0x00000000-0x0003FFFF)

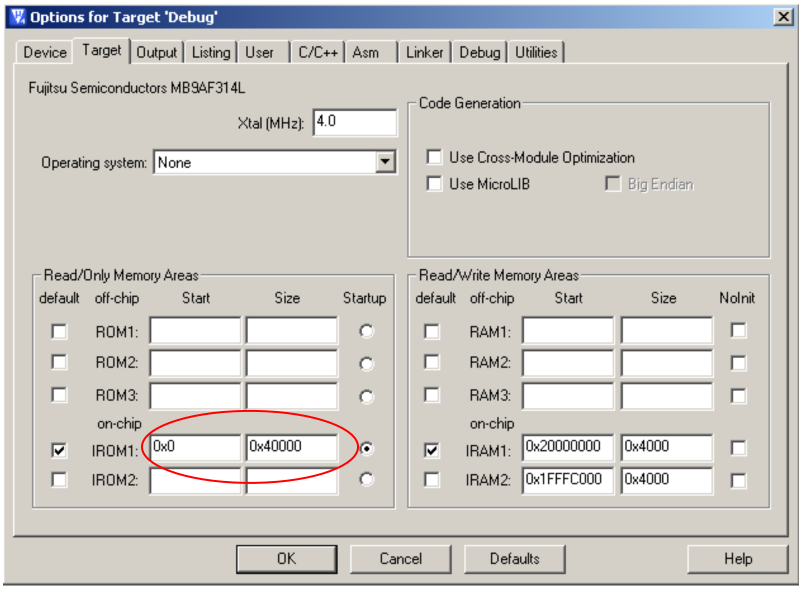


Figure 5-22: ROM Address Setting for Flash Debug

2) Don't use initialization file.

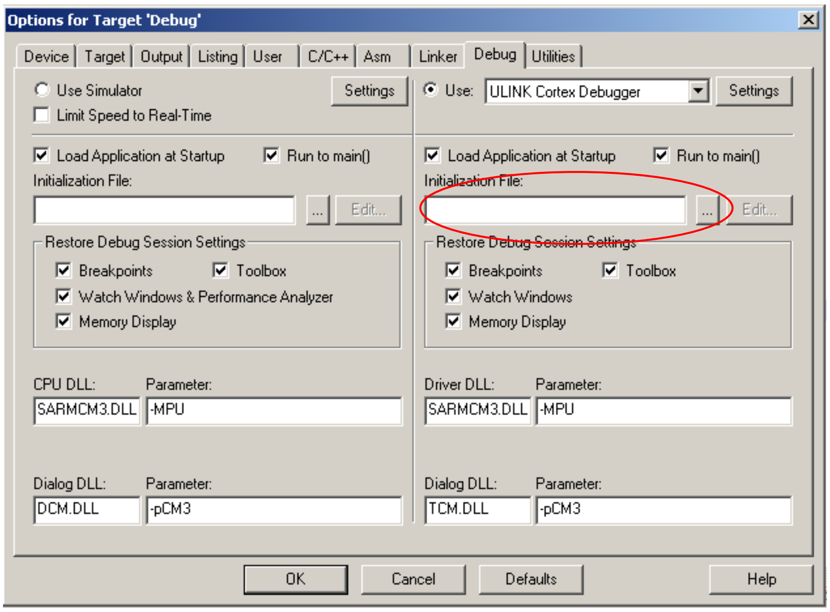


Figure 5-23: Don't use initialization File

3) Check "Update Target before Debugging" checkbox

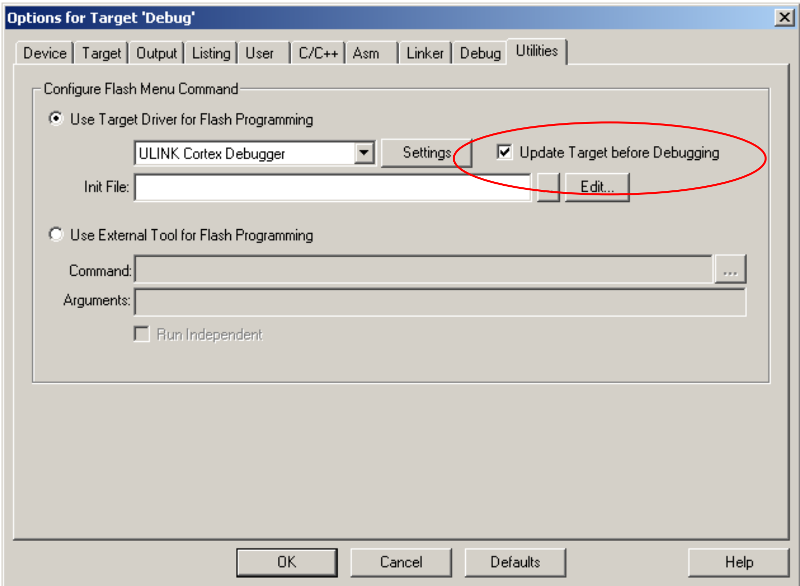


Figure 5-24: Select Update Target Before Debugging

■ Setting for RAM Debug

1) Set ROM address in Code SRAM area. (0x1FFFC000-0x1FFFFFFF)

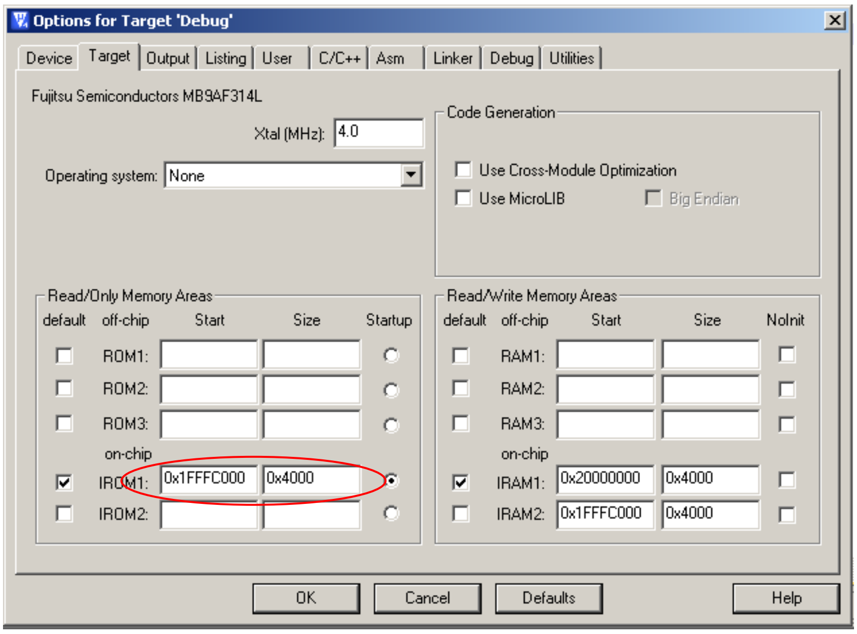


Figure 5-25: Set RAM Address for RAM Debug

2) Set initialization file path. (..\Debug_RAM.ini)

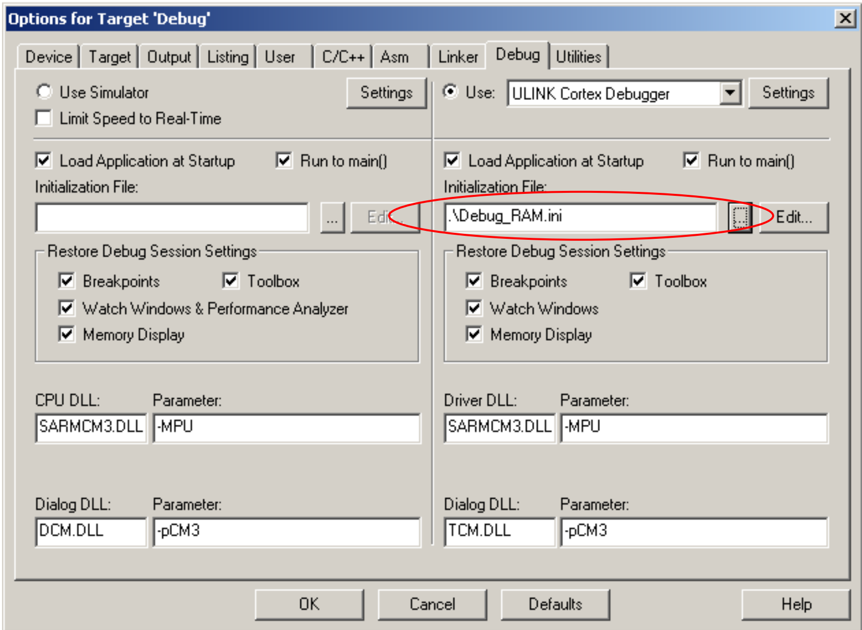


Figure 5-26: Select Initialization File

3) Don't Check "Update Target before Debugging" checkbox

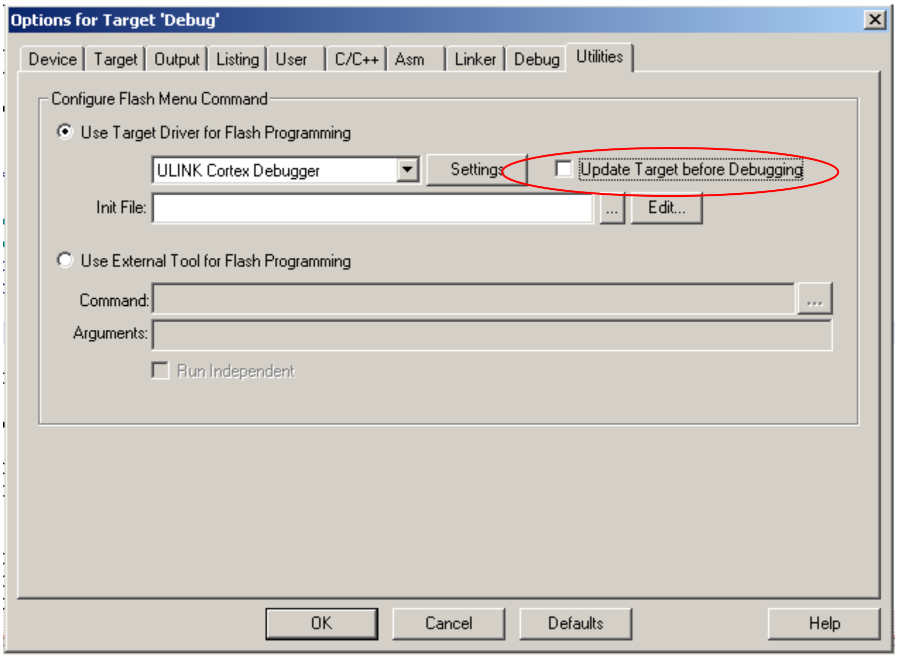


Figure 5-27: Select Update Target Before Debugging

If user need to program the hex file into Flash via USB programmer, a hex file need to be produced first.

How to Make a HEX File

1) Check "Create HEX File" checkbox (This file is Intel Format HEX)

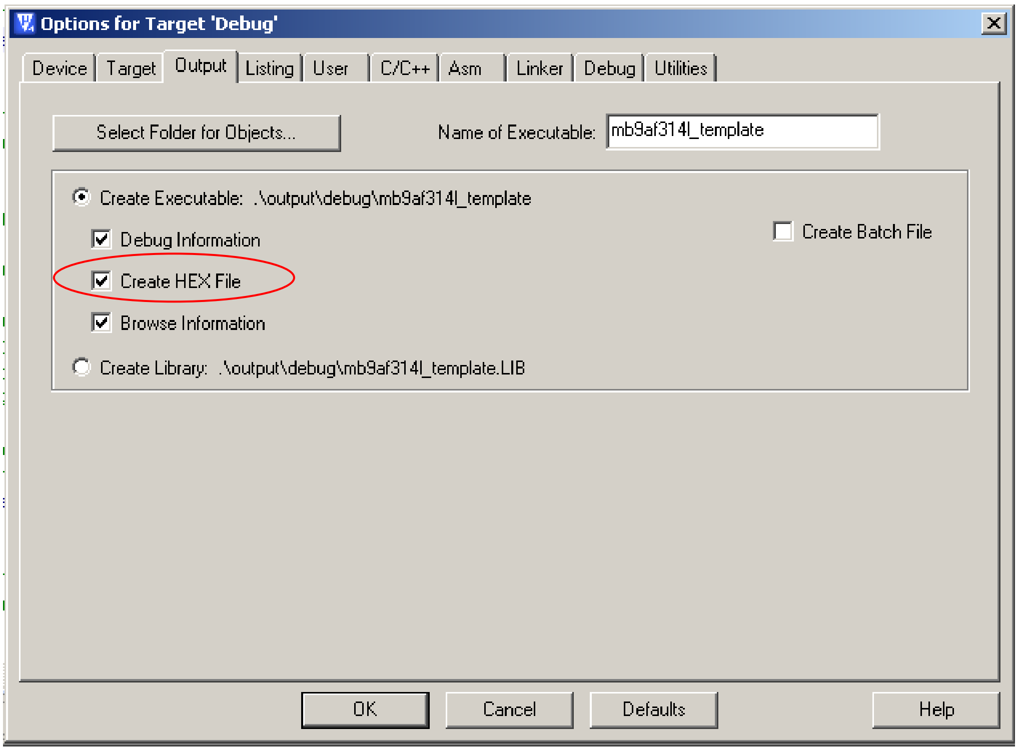


Figure 5-28: HEX File Generation in Keil IDE

2) User can find the generated file in path "..\output\debug\"

6 Materials Download

The following materials can be downloaded from below web.

http://www.fujitsu.com/cn/fss/mcu/32bit/fm3/

- **Software**
  - FUJITSU Flash MCU Programmer
  - FUJITSU Flash USB DIRECT Programmer

- **Document**
  - FSSDC-9A314-EVB User Manual
  - FSSDC-9A314-EVB Schematic
  - MB9A310 Series Datasheet
  - MB9A310 Series Peripheral Manual
  - MB9A310 Series Flash Programming Manual

- **Sample code**
  - Blinky
  - usb_func_mouse
  - usb_host_catch_mouse
  - usb_host_msc

- **IDE Study Material**
  - IAR IDE study material
  - Keil IDE study material